PATENT APPLICATION OF

MARIO BLAUM AND MANTLE YU

FOR

DISK DRIVE HAVING IMPROVED TIMING MARKS

CROSS REFERENCE TO RELATED APPLICATIONS

This application is related to copending, commonly assigned application docket number HSJ9-2003-0240US1, entitled "Method for operating disk drive having improved timing marks", filed on even date herewith, and hereby incorporated by reference.

FIELD OF THE INVENTION

This invention relates to disk drives for data storage, and in particular to disk drives that have timing marks on disk tracks for servo head control or for data synchronization.

BACKGROUND

Data is recorded on magnetic disk drives in radially spaced tracks on the surface of one or more rotating disks. Typically, a single recording head, which may be an inductive read/write head, or an inductive write head in combination with a magnetoresistive read head, is associated with a corresponding magnetic recording surface of each disk. Each recording head is moved by an actuator in a generally radial direction toward and away from the center of rotation of the disks, to align the head to a desired track.

It is necessary to know the precise radial and circumferential location of the recording heads relative to

their associated disk surfaces. For conventional fixed-block architecture disk drives, position information is typically recorded onto the disk as servo information in angularly spaced servo sectors interspersed among the data sectors.

Each of the servo sectors contains a servo timing mark (STM), which is a defined bit pattern. A servo timing mark is also known as a servo identification (SID) or a servo address mark (SAM) in the art. When an STM is identified in reading the disk, subsequent detection of servo information (e.g., track identification and position error signal bursts) is initiated. This servo information is used by servo electronics to determine the radial position of the head and to provide feedback to the actuator to ensure the head remains positioned over the centerline of the desired track. In some cases, the STMs are also used to assist in locating specific data sectors where user data is to be read or written (e.g., as described in US Patent 5,500,848).

Accurate detection of STMs is crucial to proper disk drive operation, since such detection is required in order to correctly recognize subsequent servo information. If a servo sector is not recognized due to failure to detect the STM, the servo electronics will rely on less recent servo information (e.g., from the most recent recognized servo sector), and servo tracking and timing accuracy will be diminished. Also, if an STM is incorrectly detected at the wrong location, subsequent servo information will be missed and/or incorrectly interpreted and acted upon. Incorrectly recognizing an STM and acting incorrectly as a result tends to be more problematic than failure to recognize an STM.

Detection of an STM pattern within a servo sector can be reformulated as detection of a known bit pattern (i.e., the STM bit pattern) embedded within an input stream subject to error (i.e., the sequence of bits read from the disk). A known approach for locating the STM bit pattern within the input stream is to compare an n-bit STM bit pattern to a window consisting of n consecutive bits in the input stream, and successively shifting the window, one bit at a time, until a match is found between the window and the STM bit pattern.

How well two bit sequences of equal length match each other is conveniently described by the Hamming distance, which is the number of bits which differ in the two sequences. A perfect match corresponds to a Hamming distance of zero. It is also convenient to define the above shifts as pre-shifts, where pre-shift 1 corresponds to a window position starting 1 bit before the start of the STM pattern in the input stream, pre-shift 2 corresponds to a window starting 2 bits before the STM pattern, etc. Accordingly the above procedure can be expressed as calculating the Hamming distance for pre-shifts k, k-1, k-2, …, 2, 1, 0 in succession, for a suitably chosen integer k, and identifying pre-shift 0 (i.e., the STM) when a Hamming distance of 0 is found.

In the absence of errors, a perfect match will occur between the STM pattern and the window when the window is aligned to the STM pattern in the input stream, and this perfect match indicates detection of the STM on the disk, provided there is no other perfect match to the STM pattern in the input stream. Naturally, it is necessary to restrict the search for an STM to regions of the disk which are substantially free of user data, since user data may

coincidentally contain the same bit pattern as the STM bit pattern.

In the presence of errors, a perfect match is not to be expected, even when the window is aligned to the STM pattern in the input stream. However, it is known in the art to select an STM pattern that provides generally large Hamming distance for pre-shifts 1 through n for an n-bit STM pattern. Such an STM pattern minimizes the chances of misidentification of the STM pattern based on a low Hamming distance between the STM pattern and a pre-shifted position of the window. Since the pre-shifted window includes bits from the input stream before the STM pattern, a preamble bit sequence having at least n bits is typically prepended to the STM pattern on the disk. Inclusion of the preamble fixes the pre-shifted bit patterns, and renders them independent of any data or other information on the disk. A commonly used preamble pattern is all ones, but any other bit pattern may be used as well. If the minimum Hamming distance between the STM pattern and pre-shifts 1 through n is d1, then d1 is referred to as the pre-shift sliding distance.

However, this prior art approach has some drawbacks. Essentially, a search for the presence of an STM within an STM search window on the disk is performed, since this search cannot go on indefinitely. If the STM is not found within the search window, some alternative correction action is taken. If the STM search window does not extend past the STM, then the probability of missing the STM is undesirably increased (e.g., if the STM window is set incorrectly by only 1 bit so that it does not include the entire STM, the STM will be missed in the search). However, if the STM search window extends past the STM,

then erroneous recognition of the STM is possible, because the prior art STM patterns are optimized only with respect to pre-shifts. Erroneous recognition of an STM pattern is typically a much more severe problem than not finding an STM pattern, since erroneous recognition may lead the disk drive to take further erroneous decisions (e.g., misidentifying the track).

Accordingly, it would be an advance in the art to provide an STM search window which extends past the STM on the disk while avoiding the erroneous recognition problem identified above. It would also be an advance to provide STM patterns compatible with this STM search window.

SUMMARY

In one aspect of the invention, the present invention provides a disk drive having timing marks (TMs) within timing sections of disk tracks, with reduced probability of missing or misidentifying TMs in operation. More specifically, the disk drive searches for TMs on the disk within an TM search window which extends past the TM on the disk, which reduces the probability of missing an TM due to misalignment of the TM search window with respect to the TM.

TM patterns for use with embodiments of the present invention are preferably chosen to reduce the probability of misidentification of an TM in the presence of read errors. More specifically, an n-bit TM is preferably chosen to maintain a maximal post-shift sliding distance d2 for m post-shifts of the TM pattern relative to the input stream, where m is selected to correspond to the nominal boundary of the TM search window. In this manner, the probability of a misidentification of the TM due to a post-

shift having a small Hamming distance from the TM pattern may be reduced. In addition, the TM pattern maintains a selected pre-shift sliding distance d1 for n pre-shifts of the TM pattern relative to the input stream.

In a preferred embodiment, the timing marks are servo timing marks (STMs), and are followed by servo position information. Recognition of the STMs enables the servo position information to be used to control the head position during disk operation. In an alternate preferred embodiment, the timing marks are data timing marks (DTM), and are followed by data. Recognition of the DTMs facilitates byte synchronization of data.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 schematically shows a disk drive according to one embodiment of the invention.

Figure 2a schematically shows a portion of a track from a disk drive showing a servo sector and a data sector, according to an embodiment of the invention.

Figure 2b schematically shows details of the servo sector of Figure 2a.

Figure 3 is a table showing Hamming distances from a pattern u1 = {0,0,0,1,0,0,1} to pre-shifted and post-shifted instances of u1, using a preamble pattern of all ones, according to an embodiment of the invention.

Figure 4 is a table showing Hamming distances from a pattern u2 = {0,0,1,1,1,0,1} to pre-shifted and post-shifted instances of u2, using a preamble pattern of all ones, according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE DRAWINGS

Figure 1 schematically shows a disk drive 10 that is representative of the type of disk drive according to one embodiment of the invention. Disk drive 10 includes disk 12, read/write head 16 (also called a data recording transducer), actuator 18, TM decoder 20, and servo electronics 22. Disk 12 has a set of radially spaced tracks, one of which is shown at 14. Disk 12 is rotated about its center 13 by a motor (not shown). Information is written to and/or read from track 14 by read/write head 16. Read/write head 16 is connected to actuator 18, which radially positions head 16 over a selected track. For example, actuator 18 may be rotated about axis 19 by an actuator motor (not shown).

Figure 2a schematically shows a portion of track 14. Track 14 includes servo sectors (SS) interspersed with data sectors (DS). A servo sector is shown as 30 in Figure 2a, and a data sector is shown as 32 in Figure 2a. Figure 2b shows a more detailed view of servo sector 30 of Figure 2a. Servo sector 30 includes a preamble bit pattern (P), shown as 34, a servo timing mark bit pattern (STM), shown as 36, and servo information (SI) shown as 38. In Figure 2b, the head encounters regions 34, 36, and 38 in that order (i.e., the tracks shown in Figures 2a and 2b are read from left to right).

Returning now to Figure 1, read/write head 16 is operably connected to TM decoder 20, as indicated by line 24. As track 14 passes under head 16, the head will encounter servo sectors 30 and data sectors 32. TM decoder 20 receives a bit stream corresponding to the bit stream on track 14, and functions to detect the STM bit pattern 36 within servo sectors 30 as they pass under the head. Upon identification of an STM bit pattern 36, TM decoder 20

transmits an "STM found" signal to servo electronics 22, as schematically indicated by line 26 in Figure 1. Servo electronics 22 also receives servo information 38 from TM decoder 20, and makes use of servo information 38, gated by the "STM found" signal, to perform closed loop control of actuator 18, schematically indicated by line 28, such that head 16 is centered over a desired track (i.e., track 14 in this example). Further details of head control in response to servo timing marks 36 and servo information 38 are given in US Patent 5,903,410, incorporated by reference in its entirety.

In order to reduce the probability of misidentification of an STM, TM decoder 20 usually only looks for STMs within a STM search window, which nominally extends several bits past an STM on the disk and at least n bits before an STM on the disk. In normal disk drive operation, information obtained from the previous servo sector encountered will enable the STM search window to be set with sufficient accuracy to define its nominal boundaries with respect to the next STM expected as indicated above. Unusual events, such as track changes, error recovery and disk drive start-up use other methods for bounding the search for the STMs, which are known in the art.

Typically, TM decoder 20 compares STM pattern 36 to a window of consecutive bits read from servo sector 30, and this window is successively stepped through the bit stream read from servo sector 30. For example, reading the bit stream from servo sector 30 into a shift register is one way to provide this stepped windowing.

Therefore, the performance of an STM pattern u1 can be assessed by computing Hamming distances between u1 and

shifted versions of u1, as shown on Figure 3 for u1 = {0,0,0,1,0,0,1}. The top row of Figure 3 shows the sequence of bits on the disk, also referred to as the input stream. In this example, the STM pattern u1 is preceded by a preamble pattern which is all ones. Pre-shift k, indicated by "pre k" on Figure 3, indicates a window position that is shifted k bits before the STM pattern in the input stream. Similarly, post-shift k, indicated by "post k" on Figure 3, indicates a window position that is shifted k bits after the STM pattern in the input stream.

Due to the presence of the preamble pattern in the example of Figure 3, the Hamming distances between u1 and each of the pre-shifts can be computed straightforwardly, and are given in the "distance" column of Figure 3. In this example, the smallest Hamming distance between u1 and any of pre-shift 1 through pre-shift n, also known as the pre-shift sliding distance, is 4. Here n is the number of bits in the STM pattern. When the window is aligned to the STM pattern, indicated by "align" on Figure 3, the Hamming distance between u1 and the window is zero, as indicated on Figure 3.

Figure 3 also shows the first few post-shifts. For post-shifts, the window extends past the end of the STM pattern and covers bits which are not part of the STM bit pattern. It is possible to add a postscript bit pattern after the STM, similar to the preamble bit pattern before the STM, to make the Hamming distance calculation definite for post-shifts. Alternatively, it is also possible to perform the distance calculation for post-shifts as a "worst-case" calculation, where all bits of the window extending past the STM pattern on the disk are assumed to match the STM pattern. The worst case approach has the

advantage that fewer bits on the disk are required for STM detection than if a postscript pattern is used (which increases the fraction of disk area devoted to user data), while the use of a postscript pattern will provide improved error resistance. The post-shift distances given in Example 3 are calculated using the above worst case assumption.

The STM pattern of Figure 3 provides a pre-shift sliding distance of 4, and a post-shift sliding distance of 3 for two post-shifts past alignment. Since the number of post-shifts past alignment which are of interest is a variable parameter, the post-shift sliding distance is expressed with the notation (d2; m), meaning that distance no less than d2 is obtained for the first m post-shifts past alignment. With this notation, the worst case post-shift sliding distance is the minimum distance between the first n-k bits of the STM bit pattern and the last n-k bits of the STM bit pattern as k is varied from 1 to m. We have found that the pre-shift sliding distance and the post-shift sliding distance can be traded off. For example, by reducing a requirement on pre-shift sliding distance, it is usually possible to increase m for a fixed d2 in the post-shift sliding distance (d2, m).

Figure 4 provides a demonstration of such a trade. The STM sequence in Figure 4 is u2 = {0,0,1,1,1,0,1}. As seen in Figure 4, this sequence provides a pre-shift sliding distance of 3 (one less than provided by u1 of Figure 3), and provides a post-shift sliding distance of (3; 3) as opposed to the post-shift sliding distance of (3; 2) provided by u1 of Figure 3.

STM patterns having desirable pre-shift and post-shift distances can be determined using a systematic computer-

assisted search. Results of such a search for 16-bit STM patterns are presented in the following table:

pattern	pre-shift sliding distance d1	post-shift sliding distance (d2; m)
0001 0100 1001 1101	9	(9; 1)
0001 0100 1001 1101	9	(8; 1)
0001 0100 1001 1101	9	(7; 2)
0001 0100 1001 1101	9	(6; 4)
0001 1001 0101 0011	8	(8; 3)
0010 0010 1100 0011	8	(7; 5)
0000 0111 0101 0011	8	(6; 7)
0000 0101 1011 1001	8	(5; 9)
0010 0111 0100 0011	7	(7; 6)
0000 0101 1011 1000	7	(6; 8)
0000 0101 1011 1100	7	(5; 10)
0100 0111 1101 1000	6	(6; 9)
0000 0101 1011 1100	6	(5; 10)
0000 0001 0100 1111	6	(4; 12)

Table 1: Error-resistant 16-bit STM patterns

In Table 1, the STM patterns are given in the leftmost column. Spaces are inserted into the patterns as shown in Table 1 for readability only. On the disk, these STM patterns occupy 16 consecutive bit positions. For each case (i.e., row of Table 1), the pre-shift sliding distance d1 is given in the center column, and the post-shift sliding distance (d2; m) is given in the right column. In Table 1, post-shift sliding distances are based on the worst case calculation discussed above. The patterns in Table 1 are selected to have maximal m, given n, d1, and d2. Generally, decreasing d1 and/or d2 allows this maximal m to be increased to a larger value, as shown on Table 1, and as expected from the examples of Figures 3 and 4. The results of Table 1 are obtained using a preamble pattern of all ones. If a different preamble pattern is used,

equivalent performance will be obtained (i.e., m in each row will not change), but different STM patterns will be determined by the computer search for achieving this performance.

Although the specific optimization performed to generate Table 1 is the maximization of m given d1, d2, and n, results of other optimizations may also be determined from Table 1. Two examples are as follows: for n = 16, d1 = 9, m = 2, the maximal d2 is 7; and for n = 16, d2 = 7, m = 5, the maximal d1 is 8. Alternatively, the systematic computer search used to generate Table 1 can readily be altered to maximize d1 given the other parameters, or to maximize d2 given the other parameters, or to minimize n given the other parameters as opposed to maximizing m given the other parameters. All four approaches are substantially equivalent ways to generate STM patterns in accordance with teachings of the invention. It is convenient to refer to STM patterns generated by any of the above identified optimization methods as (n, d1, d2, m) patterns.

In practice, it is expected that the most typical optimization will be the maximization of d2, given n, d1 and m. The reason for this is that n is typically fixed by high level design considerations (e.g., how much disk area to devote to the STM patterns), d1 is selected to provide the desired level of error resistance for pre-shifts, m and the boundary of the STM search window are selected together to reduce the probability of the search window boundary falling within the STM to a design value, and then d2 is maximized to provide as much post-shift error resistance as possible given the other constraints. Providing resistance to pre-shift errors tends to be more important than

providing post-shift error resistance. One reason is that post-shift error resistance is only relevant if the STM is missed, which itself is a low probability event. Accordingly, a design will typically end up with d1 > d2, although this is not required to practice the invention.

In some cases, the computer search for optimized STM patterns will provide multiple patterns satisfying the same constraint. In these cases, one can select an STM pattern from a set of (n, d1, d2, m) patterns having the same n, d1, d2, and m according to various criteria. Such criteria include, but are not limited to, maximizing the number of ones, minimizing the length of the longest run of zeros, and minimizing the number of times the distance takes on its minimal value for pre-shifts and/or post-shifts.

The examples of Figures 3 and 4, and the results of Table 1, are all based on the use of a preamble sequence of at least n bits, where all of the preamble bits are ones. Any other fixed preamble bit sequence of at least n bits is also suitable for practicing the invention. In practice, other criteria may be used to select the preamble bit pattern, such as suitability for automatic gain control.

In the examples of Figures 3 and 4, and the results of Table 1, all of the Hamming distances were computed bitwise. This method of computing distances is suitable in cases where bit errors tend to be independent. In some cases, however, bit errors are not independent (e.g., if errors tend to occur in bursts affecting some number of consecutive bits). In such cases, the use of a burst Hamming distance (instead of a bitwise Hamming distance) to select and use STM patterns in accordance with the invention is suitable. The j-bit burst Hamming distance between two bit sequences is calculated by comparing the

two sequences j bits at a time, according to the following procedure: If two j-bit subsequences are identical, add 0 to the distance. If two j-bit subsequences differ, add 1 to the distance. Repeat for each j-bit subsequence in the two sequences. If there is a last subsequence of less than j bits, treat it the same way as all the other subsequences.

Thus the j-bit burst Hamming distance between two bit sequences is the number of differences between the two sequences, where differences are counted by comparing j-bit blocks as opposed to single bits. According to this definition, the 1-bit burst Hamming distance is identical to the bitwise Hamming distance. STM patterns according to the present invention can be generated by systematic computer searching using either a bitwise Hamming distance, or a j-bit burst Hamming distance.

Although the embodiment discussed above is presented in the context of a magnetic disk drive having a specific architecture, the invention is applicable to magnetic or optical disks having other architectures and/or other servo techniques which depend on recognition of a known bit pattern on a disk.

Furthermore, although the embodiments discussed in detail above relate to locating a servo timing mark to signal the presence of servo information on a disk, the invention is applicable to locating any mark represented by a bit pattern on a disk. For example, byte synchronization is required before reading data from a disk, and therefore a known data timing mark bit pattern typically precedes the data. The apparatus and methods of the present invention are advantageously applicable for locating such data timing marks with reduced probability of error. For both

applications (servo and data), it is necessary to locate a timing mark within a timing section within a disk track. For the servo application, the timing sections are the servo sectors identified above, while for the data application, the timing sections precede the data.